

HTTP - Hypertext-Transfer-Protocol

- HTTP ist die Grundlage der Daten-Kommunikation im World-Wide-Web
- Hypertext ist strukturierter Text mit logischen Links, Hyperlinks, zwischen verschiedenen Text-Knoten
- Die Entwicklung wurde 1989 am CERN begonnen von Tim Berners-Lee
- Erste Definition von HTTP/1.1, der heute gängigen Version, stammt von 1997, Änderungen erfolgten 1999 und 2014

HTTP - Technik

- Der Nachfolge-Standard HTTP/2 wurde 2015 festgelegt und wird jetzt von größeren Web-Servern unterstützt
- HTTP fungiert als Anfrage-Antwort Protokoll (request-response) im Client-Server Modell
- Client wäre etwa ein Web-Browser, der HTML-Dateien anfordert, oder andere User-Agents wie Web-Crawler (von Suchmaschinen-Diensten), Voice-Browser, mobile Apps, Skripte etc.

HTTP-Cache

- Web-Cache kann als Reverse-Proxy vor den Web-Server geschaltet sein zur Beschleunigung und als DDoS-Schutz
- Web-Browser haben ebenfalls einen Cache und speichern Inhalte, um möglichst den Netzwerk-Traffic zu reduzieren
- Web-Proxies haben denselben Zweck, können den Verkehr außerdem filtern und die IP-Adresse des Anfragenden verschleiern

Persistente Verbindungen

- TCP – keep-alive wurde zu HTTP/1 hinzugefügt
- In HTTP/1.1 werden alle Verbindungen als persistent fortbestehend angesehen, außer wenn anders angegeben
- Kein keep-alive mehr, sondern es können mehrere Anfragen über eine einzelne Verbindung gemacht werden
- Der Timeout beim Apache-Server beträgt standardmäßig nur fünf (5) Sekunden

HTTP-Konzept

- HTTP ist ein Anwendungs-Schicht Protokoll, die Definition geht von einem darunter liegenden zuverlässigen Transport-Schicht Protokoll aus, üblicherweise TCP/IP, siehe ISO-OSI-Schichten
- Es kann auch über das unverlässliche User-Datagram-Protocol, UDP verwendet werden
- HTTP-Ressourcen sind im Netzwerk identifiziert und verortet durch Uniform-Resource-Locators, URLs
- Verwendet werden dazu die Uniform-Resource-Identifiers Schemas:: `http://` und `https://`

HTTP-Konzept

- URI und HTML-Hyperlinks bilden Hypertext-Dokumente
- HTTP/1.1 kann TCP-Verbindungen mehrfach nutzen, um Bilder, Skripte, Stylesheets usw. zu laden, nachdem die Haupt-Seite ausgeliefert wurde
- HTTP/1.1 – Kommunikation ist deswegen mit weniger Latenz behaftet, weil weniger TCP-Verbindungen langwierig aufgebaut werden müssen

Die letzte Spezifikation von 2014 umfasst:

- Message Syntax und Routing
- Semantics and Content
- Conditional Requests
- Range Requests
- Caching
- Authentication

Vorteile persistenter Verbindungen:

- Streaming statt Buffering
- Pipelining bedeutet, dass mehrere Anfragen gemacht werden können, bevor geantwortet wird
- Geringerer Ressourcen-Verbrauch durch weniger TCP-Verbindungen
- Fehler können gemeldet werden, ohne die TCP-Verbindung zu schließen
- Vorteile haben noch mehr Gewicht bei HTTPS-Verbindungen

Zustandsloses Protokoll

- HTTP ist an sich ein zustandsloses Protokoll, das heißt dass der Web-Server keine Status-Information für jeden Anwender über mehrere Anfragen hinweg zu sammeln braucht
- Einige Web-Anwendungen implementieren aber Zustände oder serverseitige Sessions beispielsweise mit HTTP-Cookies oder mit versteckten Variablen in Web-Formularen

Links

- <https://en.wikipedia.org/wiki/HTTP>
 - https://en.wikipedia.org/wiki/HTTP_persistent_connection
 - https://en.wikipedia.org/wiki/Proxy_server
 - https://en.wikipedia.org/wiki/ISO_OSI
-
- <http://creativecommons.org/licenses/by-nc-sa/4.0/>